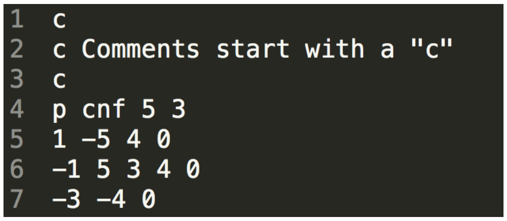
The File Format. See Figure 1 for an example.

- The file can start with comments lines that start with the character c.
- Right after the comments, you need the special line

  `p cnf $nvar $ncls`

  where `$nvar` is the total number of variables (5 here) in the file, and `$ncls` is the total number of clauses (3 here) in the file.
- Then write down the clauses. Recall that each clause represent a disjunction of literals. The variables are represented by integers starting from 1. Each clause is a sequence of non-zero integers. The 0 at the end denotes the end of the clause.
- A positive integer represents the positive occurrence of the corresponding variable (positive literal), and a negative number the negation of the variable (negative literal).



```
c
c Comments start with a "c"
c
p cnf 5 3
1 -5 4 0
-1 5 3 4 0
-3 -4 0
```

Figure 1: CNF Format

So the file in Figure 1 encodes the formula

$$(p_1 \vee \neg p_5 \vee p_4) \wedge (\neg p_1 \vee p_5 \vee p_3 \vee p_4) \wedge (\neg p_3 \vee \neg p_4).$$

Encoding Sudoku. Each spot in the grid has a coordinate (i, j) with i and j both ranging from 1 to 9 (or 0 to 8, your choice). So you need 4 Boolean variables to encode each dimension, and 8 variables can encode all spots. Then, the values that each spot can take can be represented with another 4 Boolean variables. Note that you need different Boolean variables to encode the values for different spots (but the same variables for the same spot).

For instance, suppose you want to say $x_{1,1}$ takes value 5. The corresponding encoding is

$$\Big((\neg ib_3 \wedge \neg ib_2 \wedge \neg ib_1 \wedge ib_0) \wedge (\neg jb_3 \wedge \neg jb_2 \wedge \neg jb_1 \wedge jb_0)\Big) \rightarrow \Big(\neg v_3^{11} \wedge v_2^{11} \wedge \neg v_1^{11} \wedge v_0^{11}\Big)$$

Here, the ib_k variables encode the row number (0001), and the jb_k variables encode the column number (0001). The v_k^{11} variables hold the values for this particular spot, and right now encoding the number 5 (0101). Remember that $\phi \rightarrow \psi$ is simply $\neg \phi \vee \psi$.

You first need to generate the condition that each spot has to take values from 1 to 9. Then the condition that the spots in all the rows, columns, and the 9 smaller squares must contain different values (these are constraints on the value variables). Think carefully about how to put everything into CNF (De Morgan's laws may suffice). Then, for each particular problem, add clauses that fix the values on spots whose numbers are already given.